

鋼琴智慧輔導系統

107703008 資科三 蔡桂毓 指導老師：蔡子傑老師

一、緣起 & 動機

智慧輔導系統 (ITS) 最早的概念是自 1970 年代起的電腦輔助教學 (CAI)，而初期的 ITS 系統創造是為能教導且執行特定項目並動態適應各式情況。到了現代的 ITS 系統嘗試複製老師或助教的角色並執行自動化教學，如提出問題、選擇問題和提供回饋。

而此次所建立的鋼琴智能輔導系統 (ITSP) 其注重的重點是互動式學習，在我的構想中，我所需要的將不是以往所見的 ITS 那般複雜，而是摒除現在 ITS 如教授那般所做的智慧型授課，比較像是一位家教針對不同難度的學生所提供的訓練，課程由人為修訂而輔導學生則是系統來完成，而這些互動學習包括系統給予的分數以及在學習過程中捕獲學生的行為及回饋。

二、目標完成 & 挑戰解決

此專題的最終目標是希望系統能像一位家教針對不同難度的學生所提供的鋼琴訓練，以及包含人為修訂的課程及輔導學生學習過程的系統，並且希望課程能以遊戲方式會更吸引學生來學習，做得像現今在市面流傳許多的鋼琴演奏遊戲，並加入更多的音樂知識教學用以精進樂器技巧。

而這次實作的最終結果約有達到預期成果的 70% 左右，對於演算法所建立的教學評分系統於我或是其他使用者來說都有不錯的反饋，評價也多是正面也讓一些使用者有加深他們對於鋼琴的遊戲興趣。而所缺少的 30% 於我自己評估是在上學期末報告中本有提及的基因演算法未能實作於此系統中，基因演算法可以讓此專題更加符合數據分析下的課程選取，但在碰上其他的問題下本該花費較多時間完成的演算法因而無法成功實作。再來還有對於課程的難度拿捏不太恰當經郭多位使用者實測下發現課程難度有些偏高，對於一般學童來說所需花費超過半年的學習時間與此系統僅濃縮成 20 分鐘，對於學習的實際成效沒有比起激發興趣的成效來得明顯。

而在專題中所碰到的挑戰也就是此系統的輸入，是透過電腦外接 MIDI 鍵盤來做鋼琴教學的主要輸入。在一開始所使用 windows 系統無法順利於發出聲音的同時執行此程式，因問題不得而解，在一次嘗試 macOS 時卻意外解決，故此系統目前也只能適用於 macOS 系統中。輸出則是以課程遊戲介面、琴譜及教學提示等所組成；還有背後演算法的運作方式，來讓系統能順利偵測學生學習情況並給予相應幫助，並安排合適課程提供給學生學習。

三、相關研究 & 技術

(一) 系統架構

- The Domain model

作為對於各式學生所需要的不同難度課程，以拜爾鋼琴教本作為系統的專業知識資料庫，且提供錯誤訂正的標準。

在這裡特定子集可以分成不同分數及難度的課程以區分難易度，而樂理樂句等等皆會使用到的樂器知識可被定義在定義域中。

- The Student model

與原先 ITS 相同，追蹤學生認知和情感狀態及其隨著學習演變的過程。同樣地透過我所建立之演算法做 Model Tracing 來檢視學生是否偏離 Domain（意即偏離我們所設定的教材課程(特定子集)），並提供給 Tutoring model 特定結果以提供分數來供系統做回饋。

- The Tutoring model

一樣接受來自 Domain 和 Student 的資訊，並選擇爾後課程的進度並以演算法規劃方向。系統同時也會以演算法識別學生何時偏離 Domain，為學生提供及時反饋以修正後續課程排程。

- The User Interface model

以遊戲方式會更吸引學生來學習，並加入更多的音樂知識教學並且透過遊戲精進鋼琴演奏知識及技巧。

(二) 評分機制

- Elo Rating Algorithm

Elo Rating System 是指由匈牙利裔美國物理學家 Arpad Elo 創建的一個衡量各類對弈活動水準的評價方法，是當今對弈水準評估公認的權威標準。

將課程於此算法中作為學生的對手，每個課程分別有不同的分數（難度分數），當學生選擇起始難度時將會分配相應分數（學生程度分數）。在課程進行時以音符正確與否來獲得相應分數（課程分數），算出新的分數後將分數放回前述 genetic algorithm 之 fitness function 的分數已進行至下一代。

- *Elo Rating Algorithm

init student_score (學生的**初始分數(RA)**，根據難度不同)

init course_score (課程的**初始分數(RB)**，根據難度不同)

(一開始會相同)

While n < 1000 (課程難度1000的課程為最終課程)

elo_rate(course_score, student_score) (根據學生及課程分數由elo判斷**得分率(EB)**，得分率越高，相同分數所能獲得elo分數會變高)

user_score(**SA**) = INPUT (學生於此課程獲得的分數)

result(**R'**) = course_score(**RB**) + elo_rate(**EB**) * user_score(**SA**) * 2(**K**) (根據條件算出elo分數)

if result > course_score + 80 (算出知的elo分數**高過及格標準80**)

(如課程難度為100，及格分數就是100 + 80 = 180)

student_score(**RA'**) = result (因為及格就可以往下階段課程前進，分數計算調整)

course_score(**RB'**) = next_course_score (因為及格，所以可前往下階段課程)

else (不及格)

student_score = course_score (因為不及格，故回到同一課程，並將分數回歸到課程)

course_score = result (將所獲得之elo分數當作課程分數)

(代表課程分數會變高，讓接下來課程難度不變情況下，獲得elo分數變高)

(進行第二次同課程時所獲得分數變高，**學了第二次更容易通過**)

elo_rate(course_score, student_score)(反覆計算**得分率(EB)**並作課程至達成1000分，即最後一次的課程分數)

因為只上1次課程即通過
依**分數高低判定目前程度**
得分率高較不易卡關

因為**上過2次課程**可能比
只上1次得到高分學到更多

(三) 演算法

- Longest Common Subsequence Algorithm

最長公共子序列 (LCS) 是一個在一個序列集合中用來尋找所有序列中最長子序列的問題。最長公共子序列問題是一個經典的電腦科學問題，也是資料比較程式和生物資訊學應用的基礎。

將課程後面練習以此算法作為基礎比對，因一般程式比對會以絕對位置來定義所碰到的音符，但這樣若誤觸或是因其他因素可能多或少鍵會讓判斷變得更難以預測，為了讓判斷更為公正故選用此算法來鞋著成是做基本分析及判斷。

四、 方法 & 工作項目

使用 JavaScript、HTML 與 CSS 完成 ITSP。

(一) 輸入

Web MIDI API 為 JS 的 API，對於使用 MIDI 設備創建一個可以響應樂器相關輸入的 Web 應用，讓使用者可透過外接 MIDI 鍵盤來對網頁做控制。

(二) 輸出

- 介面

透過 JS 來開發 HTML 教學平台。

- 琴譜

透過 MuseScore 譜寫書中片段做為教材，並將輸入與教材進行比對。

- 教材

透過拜爾鋼琴教本，並分科輸入程式用以系統化確立學生之鋼琴素養。

(三) 演算法

- Longest Common Subsequence Algorithm

以 Dynamic Programming 搭配基礎陣列以符合背景演算法需求。

- Elo Rating Algorithm

改寫算法中之特徵方程式加上特定分數以符合背景演算法需求。

五、 結果

這次專題我將目標族群定在約是 10 歲以下的學童及學齡前兒童來使用，並且以自身過往有長期學習的鋼琴來做為目標，而 MIDI 鍵盤及電腦相對來說硬體門檻較低。透過此系統，希望能如同先前其他 ITS 在其他領域的發展，都能對教師的教學及學生的學習產生正面的影響，同時能提升對於鋼琴學習的遊戲性及成就感。而結果論來說此專題目前還無法對於鋼琴學習有太多幫助，因為課程於此編排下難度過高，故應須重新編排課程以符合所需之要求。但對於激發鋼琴學習興趣有顯著的效果，雖然課程難度偏高但即便如此使用者仍然躍躍欲試，故成果仍在接受範圍內。